Ollscoil na hÉireann, Gaillimh ***GX_____***

National University of Ireland, Galway

Semester I Examinations 2018/ 2019

Exam Code(s)	4BCT, 1MDM, 4BS
Exam(s)	B.Sc. in Computer Science & Information Technology M.A. in Digital Media B.Sc.
Module Code(s)	CT404, CT336
Module(s)	Graphics & Image Processing
Paper No.	1
Repeat Paper	
External Examiner(s)	Dr. Jacob Howe
Internal Examiner(s)	*Dr. Sam Redfern Prof. Michael Madden

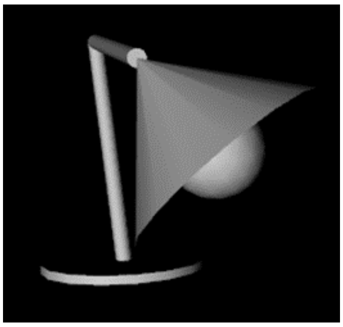
Instructions: Answer any 3 questions.
All questions carry equal marks.

Duration	2 hours
No. of Pages	6
Department(s)	Information Technology
Course Co-ordinator(s)	Dr. Des Chambers

Requirements:

MCQ
Handout
Statistical Tables
Graph Paper
Log Graph Paper
Other Material

Q.1. (Graphics)



(i) Explain, with specific reference to the depicted desk lamp model, why nested transformations are an important concept in computer graphics and animation. [5]

(ii) Provide appropriate code to illustrate the use of nested transformations in both X3D and HTML5/Canvas. [5]

(iii) The code below provides the outline of an HTML5/Canvas webpage which handles mouse down events. Write additional code (and indicate where it should be added) so that a randomly coloured square is drawn at the location of each mouse click. [5]

(iv) Indicate how you could modify the code so that each box disappears 5 seconds after it has been created [5]

```
<html>
 <head>
  <script>

function attachEvents() {
    document.onmousedown = function(event) {
        // to do: add a randomly-coloured box at the mouse position
        // note:  event.clientX and event.clientY indicate the mouse
position
    }
}

function draw() {
  var canvas = document.getElementById("canvas");
  var context = canvas.getContext('2d');

  // over-write previous content, with a grey rectangle
  context.fillStyle="#DDDDDD";
  context.fillRect(0,0,600,600);

  // to do: draw main content

  // do it all again in 1/30th of a second
  window.setTimeout("draw();",1000/30);
}
  </script>
 </head>

 <body onload="attachEvents(); draw();">
   <canvas id="canvas" width="600" height="600"></canvas>
 </body>
</html>
```

Q.2. (Graphics)

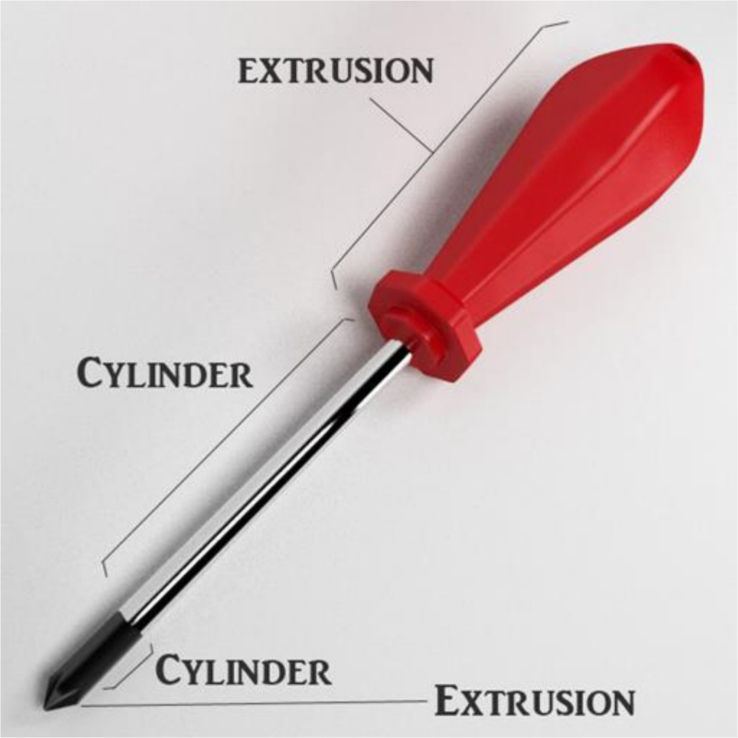
(i) With the aid of diagrams, explain the meaning of parallel projection and perspective projection, as they relate to 3D graphics. [5]

(ii) Using pseudocode, outline the key algorithmic steps involved in displaying a 3D polygonal shape which has arbitrary rotation and transformation applied. Your steps should deal with: rotation, translation, and lighting, as well as including step(s) to ensure that further-away polygons do not obscure closer ones. Note any assumptions made. [9]

(iii) Two powerful techniques for rendering shadows in realtime 3D environments are radiosity, and ambient occlusion. Explain these techniques, drawing attention to their suitability for pre-runtime computation. [6]

Q.3. (Graphics)

(i) Describe the use of extrusion in X3D, referring to each of the seven fields used by the Extrusion node. Note that extrusion and other useful nodes from the X3D language are summarised on the final page of this exam paper. [5]



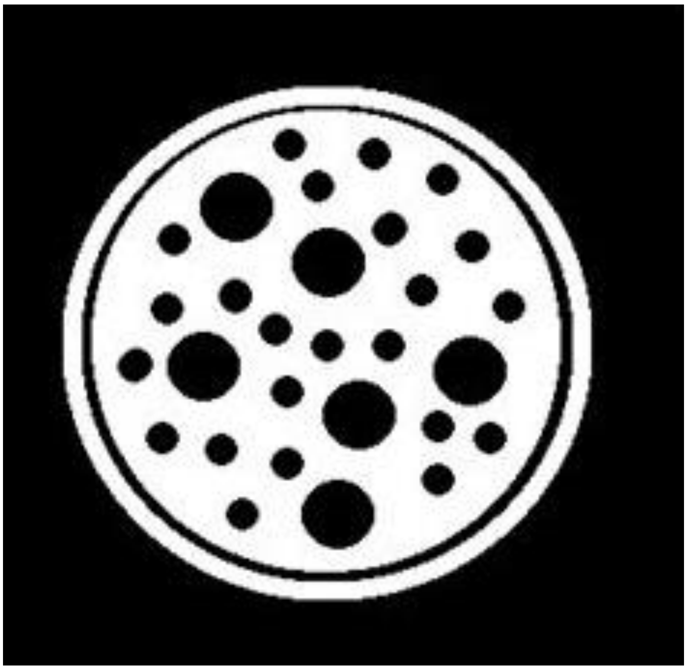
(ii) The model pictured here is of a screwdriver, which has been created using two cylinders and two extrusions. Write X3D code to create an object similar to this. You should approximate the materials as well as the geometry. **You can use a simple cross-shaped cross-section for the extruded tip of the screwdriver.** [10]

(iii) Explain how you could modify this X3D code in order to animate the screwdriver, so that it rotates clockwise and then anticlockwise each time it is clicked. You do not need to write the actual code; a technical explanation will suffice. [5]

Q.4. (Image Processing)

(i) With respect to morphological image processing, outline the following operations: erosion, dilation, opening, and closing, as applied to binary images. [8]

(ii) The black and white image below contains black circular holes in a white disc, and it is required that a fully automated system is developed to accurately count the number of large circles (of pixels radius 15 or greater) in images such as this. Present a suitable and robust set of image processing algorithms for this task. Explain why each step you have chosen is appropriate. [12]



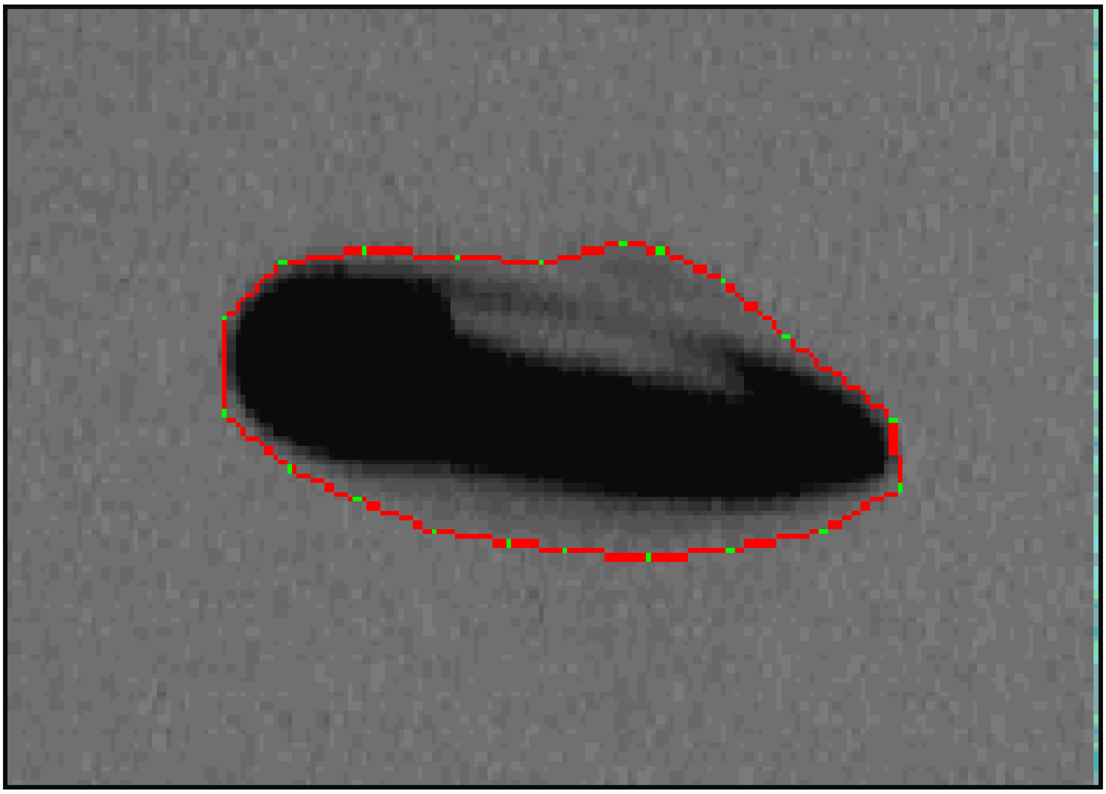
Q.5. (Image Processing)

(i) Edge Detection is an important image processing tool for identifying objects of interest in a scene. Discuss the following two edge detection techniques, indicating circumstances under which each would be useful:

- compass edge detection [4]
- boundary tracking (e.g. the Canny technique) [4]

(ii) Outline and discuss the image processing technique called active contours. Indicate why a suitable optimisation algorithm is a critical part of this technique, and briefly explain two optimisation algorithms. [7]

(iii) Present a suitable set of optimisation constraints (energy factors) for accurately tracing the outline of a bubble such as the one shown below, using active contours. [5]



Some useful X3D nodes:

Node	Important Fields and Nested Nodes
Shape	Nested Nodes: Appearance, Geometry Nodes (Box, Sphere, Cone, Cylinder, Text, Extrusion, etc.)
Appearance	Nested Nodes: Material, ImageTexture
Material	Fields: diffuseColor, specularColor, emissiveColor, ambientIntensity, transparency, shininess
ImageTexture	Fields: url
Transform	Fields: translation, rotation, scale, center. Nested Nodes: Other Transforms, Shapes, Sensors
TimeSensor	Fields: enabled, startTime, stopTime, cycleInterval, loop
PositionInterpolator	Fields: key, keyValue
OrientationInterpolator	Fields: key, keyValue
Extrusion	Fields: crossSection, spine, scale, orientation, beginCap, endCap, creaseAngle
Box	Fields: size
Sphere	Fields: radius
Cylinder	Fields: radius, height, side, top, bottom
Cone	Fields: height, bottomRadius, side, bottom
PointLight	Fields: on, location, radius, intensity, ambientIntensity, color, attenuation
ROUTE	Fields: fromNode, fromField, toNode, toField

Some useful methods/properties of the Canvas 2D Context object:

Method/Property	Arguments/Values	Notes
fillRect	(Left, Top, Width, Height)	Draw a filled rectangle
beginPath	None	Start a stroked path
moveTo	(X, Y)	Move the graphics cursor
lineTo	(X, Y)	Draw a line from graphics cursor
stroke	None	End a stroked path
fillStyle	="rgb(R,G,B)"	Set fill colour
strokeStyle	="rgb(R,G,B)"	Set line colour
save	None	Save the current coordinate system
restore	None	Restore the last saved coord system
translate	(X,Y)	Translate the coordinate system
rotate	(angle)	Rotate the coordinate system clockwise, with angle in radians
scale	(X,Y)	Scale the coordinate system independently on the X and Y axes